CLAIMS

1. (currently amended) A method for executing a target application on a host processor comprising:

interpreting each of a sequence of target instructions and executing the interpreted target instructions;

dynamically translating into host instructions each of a sequence of target instructions, wherein the translating is performed when the number of times a sequence of target instructions is executed exceeds a preset count;

storing the translated host instructions wherein a plurality of additional registers are used during execution for holding the official state of a target processor;

responding to an exception during execution of a stored translated instruction by rolling back to a previous point in execution at which correct state of a target processor is known; and

interpreting each target instruction in order from the point in execution at which a correct state of a target processor is known.

2. (previously presented) The method of Claim 1 which further comprises:

collecting statistics regarding the execution of sequences of instructions which are interpreted.

3. (currently amended) A method for executing a target application on a host processor comprising the steps of:

interpreting sequentially each of a sequence of host instructions and executing the interpreted host instructions representing each target instruction of the target application;

performing a dynamic translation process when the number of times a host instruction is executed exceeds a preset count wherein a plurality of additional registers are used during execution for holding the official state of a target processor;

responding to an exception during execution of translated host instructions representing a target instruction by returning to a previous point in execution of the target application at which a correct state of a target processor is known; and

thereafter executing host instructions by interpretation of the target instruction until the point of the exception.

4. (previously presented) A method as claimed in Claim 3 which further comprises collecting statistics regarding the execution of sequences of target instructions which are executed.

5. (previously presented) A method as claimed in Claim 4 in which the statistics include the number of times the sequence of target instructions have executed.

6. (currently amended) A method as claimed in Claim 4 in which the statistics include an address of an instruction to which a target instruction including a branch operation branches.

7. (previously presented) A method as claimed in Claim 4 in which the statistics include a likelihood of a branch being taken.

8. (currently amended) A computer-readable medium encoded with a computer program for implementing a system for executing a target application designed for execution on a target processor on a host processor having an instruction set different than that of the target processor comprising:

means for interpreting a sequence of target instructions and executing each of the interpreted target instructions;

means for dynamically translating sequences of target instructions and storing each translated sequence of instructions, wherein the translating is performed when the number of times a sequence of target instructions is executed exceeds a preset count;

means for storing each translated sequence of instructions wherein a plurality of additional registers are used during execution for holding the official state of the target processor,

means for responding to an exception during execution of a stored translated instruction by rolling back to a previous point in execution at which correct state of a target processor is known, and

means for interpreting each target instruction in order from a point in execution at which a correct state of a target processor is known through the target instruction causing the exception.

9. (previously presented) A system as claimed in Claim 8 in which the means for interpreting is an interpreter software executing on the host processor, and

the means for translating is dynamic translation software executing on the host processor.

10. (cancelled)

11. (cancelled)

12. (cancelled)

13. (cancelled)

14. (cancelled)

15. (previously presented) The method of Claim 1, further comprising:

speculatively translating target instructions into host instructions based on a likelihood of a branch being taken.

16. (previously presented) The method of Claim 1, further comprising:

collecting statistics regarding the execution of sequences of target instructions which are executed.

17. (previously presented) The method of Claim 16, wherein the statistics include the number of times a branch to target instructions have executed.

18. (previously presented) The method of Claim 17, further comprising:

speculatively translating target instructions into host instructions based on a likelihood of a branch being taken.

19. (previously presented) The system of Claim 8, further comprising:

collecting statistics including the number of times a branch of target instructions have executed.

20. (previously presented) The system of Claim 19, further comprising:

speculatively translating target instructions into host instructions based on a likelihood of a branch being taken.

21. (currently amended) A method for executing a target application on a host processor comprising:

interpreting each of a sequence of target instructions and executing the interpreted target instructions;

dynamically translating into host instructions each of a sequence of target instructions, wherein the translating is performed when the number of times a sequence of target instructions is executed exceeds a preset count;

storing the translated host instructions wherein a plurality of additional registers are used during execution for holding the official state of a target processor and wherein a buffer stores working memory state changes and official memory state changes;

responding to an exception during execution of a stored translated instruction by rolling back to a previous point in execution at which a correct state of a target processor is known; and

interpreting each target instruction in order from the point in execution at which the correct state of a target processor is known.

22. (new) A method for executing a target application on a host microprocessor, the method comprising:

interpreting each of a sequence of target operations and executing the interpreted target operations;

dynamically translating into host operations each of a sequence of target operations, wherein said translating is performed when the number of times a sequence of target operations is executed exceeds a predetermined count;

storing the translated host operations, wherein a plurality of additional registers are used during execution for holding the official state of a target microprocessor;

responding to an exception during execution of a stored translated operation by rolling back to a previous time in execution at which a correct state of a target microprocessor is known; and

interpreting each target operation in order from the time in execution at which the correct state of a target microprocessor is known.

23. (new) The method of Claim 22, further comprising:

collecting statistics regarding the execution of sequences of operations which are interpreted.

24. (new) The method of Claim 22, further comprising:

speculatively translating target operations into host instructions based on a likelihood of a branch being taken.

25. (new) The method of Claim 1, further comprising:

collecting statistics regarding the execution of sequences of target operations which are executed.

26. (new) The method of Claim 25, wherein the statistics comprise the number of times a branch of target operations have executed.

27. (new) The method of Claim 26, further comprising:

speculatively translating target operations into host operations based on a likelihood of a branch being taken.

28. (new) A computer-readable medium having stored thereon computer-executable instructions that, if executed by a system, cause the system to perform a method of executing instructions, the method comprising:

interpreting a sequence of target instructions and executing each of the interpreted target instructions;

dynamically translating sequences of target instructions and storing each translated sequence of instructions, wherein said translating is performed when the number of times a sequence of target instructions is executed exceeds a preset count;

storing each translated sequence of instructions, wherein a plurality of additional registers are used during execution for holding the official state of the target processor;

responding to an exception during execution of a stored translated instruction by rolling back to a previous point in execution at which a correct state of a target processor is known; and

interpreting each target instruction in order from a point in execution at which the correct state of a target processor is known through the target instruction causing the exception.

29. (new) The computer-readable medium of Claim 28, wherein said interpreting is performed by an interpreter software executing on a host processor, and wherein said translating is performed by dynamic translation software executing on the host processor.

30. (new) The computer-readable medium of Claim 28, wherein the method further comprises:

collecting statistics including the number of times a branch of target instructions have executed.

31. (new) The computer-readable medium of Claim 30, wherein the method further comprises:

speculatively translating target instructions into host instructions based on a likelihood of a branch being taken.